



Aluno: Lucas Perin Manchine

– Prova Bloco A



-- 1 - O quarto aluno mais novo.

```
SELECT ALUNO FROM (
    SELECT DISTINCT ON (ALUNO.ALUNO)
    ALUNO.ALUNO, timestamp FROM
    PEDIDO INNER JOIN ALUNO ON PEDIDO.ALUNO =
    ALUNO.CODIGO
    ORDER BY ALUNO,timestamp
) Q
ORDER BY timestamp desc limit 1 OFFSET 3;
```

	Data Output	Explain	Messages	Notifications
	 <b>aluno</b> character varying (64)			
1	Emílio			




-- 2 - O segundo e o terceiro alunos mais novos.

```
SELECT ALUNO FROM (
    SELECT DISTINCT ON (ALUNO.ALUNO)
    ALUNO.ALUNO, timestamp FROM
    PEDIDO INNER JOIN ALUNO ON PEDIDO.ALUNO =
    ALUNO.CODIGO
    ORDER BY ALUNO,timestamp
) Q
ORDER BY timestamp desc limit 2 OFFSET 1;
```

	 <b>aluno</b> character varying (64)	
1	Regina	
2	Cris	

-- 3 - O valor total de cada pedido realizado.

```
select
    pedido,
    sum(valor)
from
    pedido_detalle
group by
    pedido;
```

	 <b>pedido</b> integer		<b>sum</b> real	
1		3	270	
2		5	170	
3		4	600	
4		6	170	
5		2	770	
6		7	270	
7		1	600	

```
-- 4 - O valor do curso mais barato.
select valor from curso
order by valor asc limit 1;
```

	valor real
1	187

```
-- 5 - Exiba o número do pedido, nome do aluno e quantos cursos foram comprados no
-- pedido para todos os pedidos realizados que compraram três ou menos cursos.
select pedido,aluno.aluno,count(pedido_detalhe.curso) from pedido
inner join ALUNO on pedido.aluno = aluno.codigo
inner join PEDIDO_DETALHE on pedido.cod = pedido_detalhe.pedido
group by pedido,aluno.aluno
having count(pedido_detalhe.curso)<4;
```

	pedido integer	aluno character varying (64)	count bigint
1	1	Wagner	2
2	3	Emilio	1
3	2	Wagner	3
4	5	Regina	1
5	7	Fernando	1
6	4	Cris	2
7	6	Fernando	1

```
-- Prova Bloco B
```

```
-- 6 - Crie uma Trigger que permita verificar o momento de inserção de dados na tabela curso.
```

```
CREATE TABLE curso_auditoria(
    codigo_curso serial NOT NULL,
    data_alteracao TEXT NOT NULL
);
```

```
select * from curso;
select * from curso_auditoria;
```

```
CREATE OR REPLACE FUNCTION curso_log_func()
RETURNS trigger AS $teste_trigger$
BEGIN
INSERT INTO curso_auditoria (data_alteracao) VALUES (current_timestamp);
RETURN NEW;
END;
$teste_trigger$
LANGUAGE plpgsql
```

```
CREATE TRIGGER log_trigger
AFTER INSERT ON curso
FOR EACH ROW
EXECUTE PROCEDURE curso_log_func();
```

-- 7 - Demonstre que sua Trigger da questão anterior está funcionando com uma inserção de um novo aluno.

```
INSERT INTO curso(codigo,curso,tipo,instrutor,valor)
VALUES
(5, 'Python para Jedis', 2, 2, 500)
```

```
select * from curso_auditoria;
```

	<b>codigo_curso</b> integer	<b>data_alteracao</b> text
1	1	2020-06-23 22:17:28....

-- Prova Bloco C

-- a) Total de Gols de uma seleção diferente da Seleção Brasileira.

-- b) Estádios dos Jogos da Seleção Brasileira.

-- letra b

```
select estadio,B.sigla from jogo
inner join equipe_pais_ano B
on
    B.cod_equipe = jogo.cod_equipe_a
    or
    B.cod_equipe = jogo.cod_equipe_b
GROUP BY estadio,B.sigla
HAVING B.sigla='BRA';
```

	<b>estadio</b> character varying (50)	<b>sigla</b> character varying (3)
1	Ellis Park Stadium	BRA
2	Moses Mabhida Stadium	BRA
3	Soccer City	BRA